

효율적인 경로 탐색을 위한 A*와 유전자 알고리즘의 결합

김광백*

Combining A* and Genetic Algorithm for Efficient Path Search

Kwang Baek Kim*

Division of Computer Software Engineering, Silla University, Busan 46958, Korea

요 약

본 논문에서는 A*와 유전자 알고리즘을 적용하여 최적 경로를 탐색하는 방법을 제안한다. 최적 경로를 탐색하기 위해 A* 알고리즘을 적용하여 출발지 노드로부터 중간 경로 노드까지의 거리를 측정하여 개체를 생성한다. 적합도 함수를 이용하여 최적 출력 노드들을 탐색하기 위한 적합도를 계산한다. 적합도에 따라 교배할 노드와 교배 지점을 선택한다. 선택한 노드와 교배 지점을 기반으로 개체들을 교배한다. 교배를 통해 새로운 개체를 생성하고 새로운 개체가 적합도 조건에 만족하면 출력 노드로 선택하고, 다음 출력 노드를 선택하기 위한 출발 지점의 노드로 설정한다. 이러한 과정들을 반복하여 최적화된 출력 노드를 도출한다. 제안된 방법을 경로 탐색 문제를 대상으로 실험한 결과, A* 알고리즘만을 이용한 경우보다 제안된 방법이 경로 탐색 문제에 있어서 최적화된 거리를 기반으로 경로를 탐색하는 것을 확인하였다.

ABSTRACT

In this paper, we propose a hybrid approach of combining A* and Genetic algorithm in the path search problem. In A*, the cost from a start node to the intermediate node is optimized in principle but the path from that intermediate node to the goal node is generated and tested based on the cumulated cost and the next node in a priority queue is chosen to be tested. In that process, we adopt the genetic algorithm principle in that the group of nodes to generate the next node from an intermediate node is tested by its fitness function. Top two nodes are selected to use crossover or mutation operation to generate the next generation. If generated nodes are qualified, those nodes are inserted to the priority queue. The proposed method is compared with the original sequential selection and the random selection of the next searching path in A* algorithm and the result verifies the superiority of the proposed method.

키워드 : A*, 유전자 알고리즘, 최적화, 경로 탐색, 적합도

Keywords : A*, Genetic algorithm, Optimization, Path search, Fitness function

Received 24 May 2018, Revised 24 May 2018, Accepted 28 May 2018

* Corresponding Author Kwang Baek Kim(E-mail:gbkim@silla.ac.kr, Tel:+82-51-999-5052)

Division of Computer Software Engineering, Silla University, Busan 46958, Korea

Open Access <http://doi.org/10.6109/jkiice.2018.22.7.943>

pISSN:2234-4772

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

정보 공학에서 최적(화)란, 사용 중인 시스템을 변경하거나 수정하여 효율적인 작업을 도출하거나 자원을 효율적으로 사용할 수 있도록 하는 작업을 의미한다. 이를테면, 컴퓨터 소프트웨어 실행 속도를 향상시키거나 기억 장치나 자원을 효율적으로 운영하거나, 전력 소모량을 적게 함으로써 최적화할 수 있다[1]. 소프트웨어에서 사용되는 계산 작업은 여러 방식에 따라 효율성이 달라진다. 예를 들어 1 ~ N 까지의 합을 계산할 때 1 + 2 + 3 + ... + N을 하는 경우(방법 1)와 수학 공식인 $(N + (N + 1)) / 2$ 를 사용하여 계산(방법 2)할 수 있다. 둘 다 기능이 똑같이 수행되지만 계산하는 방법 중, 더 효과적인 방식을 채택하는 것이 최적화이다. 최적화는 언제나 명백하거나 직관적인 과정은 아니다. '최적화된' 버전은 원래의 소프트웨어보다 실제로 실행 시간이 더 소요될 수도 있다. 이를테면 N이 작고 컴퓨터가 나누기, 곱셈보다 더하기, 루프 함수를 수행하는 부분이 더 빠르다면 방법 2보다 방법 1이 더 효과적이다[2, 3].

A* 알고리즘은 타 경로와의 비교 없이 목적지까지 탐색하기 때문에 탐색 소요 시간이 적게 소요되는 장점이 있다. 그러나 A* 알고리즘은 일반화된 규칙성이 없어서 복잡한 구조를 가지는 탐색 문제에서는 최단경로를 탐색할 가능성이 낮다. 그리고 A* 알고리즘은 노드와 노드 사이의 최적화된 거리를 계산해주는 알고리즘이기 때문에 여러 노드에 대해 최적의 경로를 도출할 수 없다. 그 이유는 경험적인 특성에 따라서 최단 경로가 아닌 근사 경로를 탐색하거나 탐색되지 않았던 지점으로 다시 탐색을 반복하는 경우에는 탐색 시간이 많이 소요된다.

따라서 본 논문에서는 단순히 A* 알고리즘을 적용하여 순차적으로 경로를 탐색하였을 때와 제안된 경로 탐색 방법을 적용하여 최적의 경로를 탐색하였을 때를 비교한다. 제안된 경로 탐색 방법에는 최적화 방법 중의 하나인 유전자 알고리즘을 적용하여 모든 노드들을 최적으로 탐색하도록 한다. 출발지 노드에서 중간 노드들의 거리를 A* 알고리즘으로 계산한 후, 계산된 거리값을 이용하여 개체를 생성한다. 생성된 개체를 유전자 알고리즘의 선택 연산과 교차 연산을 이용하여 새로운 개체를 생성한다. 생성된 개체를 적합도 조건에 적용하여 최적의 경로를 탐색한다.

II. 제안된 경로 탐색 방법

A* 알고리즘은 범용적인 경로 탐색 문제를 해결하기 위해서 널리 사용되는 방법이다. 이 알고리즘은 장애물을 피하면서 정확하지 않은 경로를 탐색하는 것을 최소화 하고 주어진 출발지와 최종 목적지까지 가는 최단 경로를 탐색하는 방법이다. 그러나 A* 알고리즘은 노드들을 재 방문해야하는 단점이 있고 여러 노드에 대해 최적의 경로를 도출할 수 없다.

따라서 본 논문에서는 A*와 유전자 알고리즘을 이용하여 모든 대상 노드를 탐색하여 최적의 경로를 탐색하는 방법을 제안한다. 최적의 경로 탐색 방법의 순서와 개요도는 그림 1과 같다.

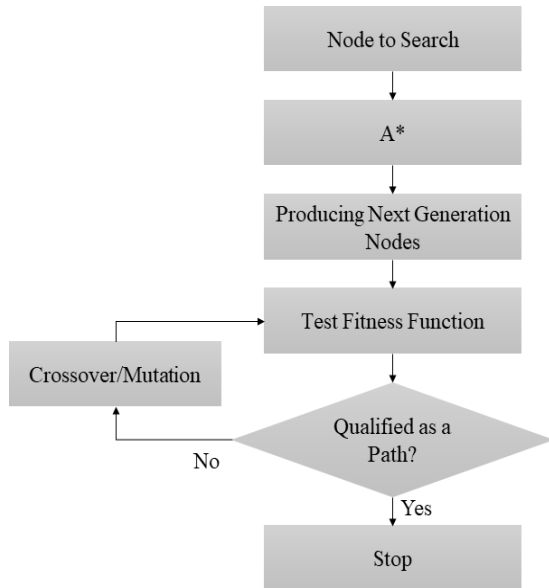


Fig. 1 Proposed Next Node Selection Method

III. A* 알고리즘을 이용한 노드간의 거리값 계산

유전자 알고리즘을 적용하기 위해 각 노드 간에 거리를 A* 알고리즘을 이용하여 계산한다. 이때 각 노드간 거리를 계산할 때에는 식 (1)을 적용하여 거리값을 계산한다.

$$f(n) = g(n) + h(n) \tag{1}$$

식(1)에서 n 은 경로의 마지막 노드이고 $g(n)$ 은 출발 꼭짓점으로부터 꼭짓점 n 까지의 경로 비용이며 $h(n)$ 은 n 에서 목표 꼭짓점까지의 추정 경로 가중치를 의미한다. A* 알고리즘에는 열린 목록(openlist)과 닫힌 목록(closetlist)이 사용된다. A* 알고리즘은 다음과 같다[4].

Step 1. 시작 노드를 열린 목록에 추가한다.

Step 2. 다음을 반복한다.

Step 2-1. 열린 목록에서 가장 낮은 $F(n)$ 비용을 선택한다.

Step 2-2. Step2-1에서 선택된 노드를 닫힌 목록으로 전환한다.

Step 2-3. 선택된 노드의 주위 8방향 노드에 대해

- 갈 수 없는 경로이거나 닫힌 목록에 있을 경우에는 무시한다.
- 열린 목록에 없는 노드일 경우, 열린 목록에 추가한 후, 선택된 노드를 부모 노드로 설정한다.
- 열린 목록에 있는 경우에는 $g(n)$ 비용 값을 사용하여 해당 노드에서 부모 노드보다 낮은 값이 있는지 확인 후, 낮은 값이 있다면 부모노드를 변경하고 $f(n)$ 값을 재계산한다.

Step 3. 종료 조건을 검증한다.

Step 3-1. 도착 노드가 열린 목록에 추가되었을 때 (경로를 찾았을 경우) 종료한다.

Step 3-1. 열린 목록이 비었을 경우(경로가 없을 경우)에도 종료한다.

IV. 유전자 알고리즘을 적용한 최적화 경로 탐색

유전자 알고리즘은 적자생존의 원리와 유전 법칙을 기반으로 제안된 최적화 방법으로 선택, 교배, 돌연변이 등과 같은 연산으로 구성된다. 유전자 알고리즘은 성능이 우수하기 때문에 변수와 제약이 많고 다수의 조건을 갖는 복잡한 문제에 널리 적용되고 있다. 유전자 알고리즘은 다양한 분야에서 최적의 해 집단을 탐색하는데 널리 적용되고 있지만 문제가 계산 불가능할 정도로 지나치게 복잡할 경우에는 유전자 알고리즘을 적용하여, 실제 최적해를 구하지는 못하더라도 최적해에 가까운 해를 구하기 위한 방안으로써 접근할 수 있다. 이 경우에 해당 문제를 해결하는 데 최적화되어 있는 알고리즘보다 좋은 성능을 보여주지는 못하지만, 대부분 받아들일

수 있는 수준의 해를 보여줄 수 있다[5].

본 논문에서의 유전자 알고리즘은 선택, 교차, 변이 연산으로 구성된다. 유전자 알고리즘의 순서도는 그림 2와 같다.

4.1. 개체생성

최적의 경로를 탐색하기 위해 각 노드간의 거리를 가중치 값으로 계산하여 유전자 알고리즘의 개체를 생성한다. 개체의 생성은 식(1)을 이용하여 가중치 값에 해당하는 개체를 생성한다. 생성된 개체들에 대해 다음 세대 개체들의 적합도 평가 기준은 식(2)와 같이 계산한다.

$$f(x) = O_{Max} * \alpha \quad (2)$$

식(2)의 O_{Max} 는 생성된 개체들의 값 중에서 최대값이고, α 는 적합도 평가 함수의 파라미터로서 초기에는 2.0으로 설정한다. 개체가 적합도의 평가 기준을 만족하지 못한 상태에서 제안된 경로 탐색 방법을 반복하면 적합도의 기준이 감소한다. 이러한 문제점을 개선하기 위해 본 논문에서는 α 값을 0.1씩 감소시켜 적합도의 평가 기준을 설정한다.

4.2. 개체선택

유전자 알고리즘에서 선택의 문제는 유전자 알고리즘의 성능과 밀접한 관계가 있다. 어떤 선택 방법을 적용하는가에 따라 최적해로 접근하는 속도가 많이 소요되거나, 지역 최소화에 위치할 가능성이 높아질 수 있다. 또는 우수한 해를 보유한 나쁜 인자가 전체 객체에 퍼지거나, 반대로 나쁜 해를 보유한 우수한 인자가 영구히 사장될 수도 있다. 따라서 선택하는 해에 따라서 유전 알고리즘의 성능에 큰 영향을 줄 수 있다. 일반적으로는 가장 좋은 해의 순으로 선택될 확률을 높게 부여하는 방법이 널리 사용되고 있으며 추가적으로 유전 알고리즘의 성능을 향상시키는 방법으로 엘리트주의(elitism)가 있다. 따라서 제안된 방법에서는 적합도가 가장 좋은 염색체가 선택되도록 확률을 가장 높게 설정하고 확률적으로 염색체를 선택하는 방법을 이용한다. 그리고 적합도가 가장 높은 상위 2개의 염색체를 보존한다.

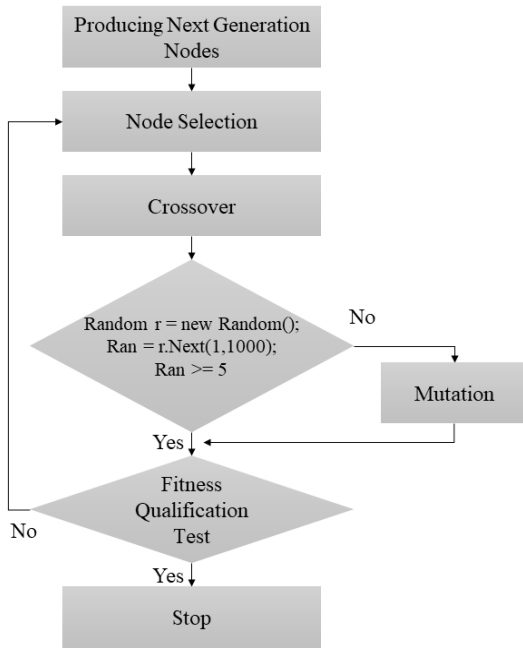


Fig. 2 Flow of Generic Algorithm Part in the Proposed Method

제안된 방법에서는 개체들의 값이 증가하여도 최적의 해를 구하도록 하기 식(3)을 개체들에 적용하여 적합도를 계산한다.

$$r(x) = O_{Max}^2 - x \quad (3)$$

식(3)의 O_{Max} 는 생성된 개체들의 값 중에서 최대값이고, x 는 개체들의 값이다. 식(3)을 이용하여 계산된 값들 중에서 적합도가 가장 높은 상위 2개의 염색체를 보존하고 다음 세대를 생산할 것인지, 아니면 세대를 생성하지 않고 도태될 것인지에 대한 기준은 식(4)를 적용하여 확률적으로 결정한다.

$$p(x) = r(x) / TN \quad (4)$$

식(4)에서 TN은 식(3)을 계산한 $r(x)$ 값들의 총합이다. $r(x) / TN$ 의 값에 반올림하여 다음 세대를 생산할 수 있는 개체와 도태되어질 개체를 확률적으로 선택한다.

4.3. 교차 연산

개체는 세대 내에서의 교배를 통해 다음 세대를 생성한다. 교차 연산에서는 두 개의 개체를 교배하여 다음 세대의 개체를 생성한다. 그리고 새로운 개체는 각각의 부모 개체로부터 서로 중복되지 않는 위치의 유전자를 받아 새로운 유전자를 구성한다. 유전 알고리즘의 교차 연산은 부모 해의 유전자들을 서로 교차시켜서 자식 해를 생성한다. 그러나 모든 해에 대해 교차 연산을 수행하는 과정에서 우수한 해가 다른 해와의 교배를 통해서 우수성이 소멸되어 세대의 최적의 우수한 해를 구할 수 없는 경우가 발생할 수 있다. 따라서 제안된 방법에서는 모든 해에 대해 교차 연산을 수행하지 않고 개체의 2/3 이후지점에 대해서만 교차 연산을 실행한다.

4.4. 유전자 변이

변이 연산은 주어진 해에서 유전자 내의 인자 순서를 바꾸어서 다른 해로 변형하는 연산이다. 따라서 약간의 확률로 변이 연산을 수행하는 것은 전체 세대가 함께 지역 최소 해에 위치할 가능성을 감소시킬 수 있는 주요한 기법이 될 수 있으며, 집단에서 잃어버린 유전 형질을 복구하여 다양성을 유지하기 위한 수단이 될 수 있다. 이때 전형적인 돌연변이 확률은 0.05 이하이다[6, 7]. 제안된 방법에서는 0.05% 확률로 변이 연산을 수행하여 지역 최적해에 위치하는 경우를 방지한다.

V. 실험 및 결과 분석

본 논문에서는 Intel(R) Core(TM) i5 CPU @ 2.80GHz 와 8.00GB RAM이 장착된 PC상에서 Visual Studio 2013 C#으로 제안된 방법을 구현하여 실험하였다.

본 논문에서 제안한 방법은 출발지 노드 및 중간 노드의 거리 값을 A* 알고리즘을 적용하여 거리 값을 계산하였다. 그리고 유전자 알고리즘에 이용하여 최적의 경로를 탐색하였다. 제안된 방법은 A* 알고리즘으로 계산된 거리 값을 이용하여 개체를 생성한 후, 선택 연산을 적용하여 적합도를 구하고 교배 연산에 적용하여 새로운 개체를 생성하였다. 새로 생성된 개체들을 적합도 평가 함수에 적용하여 적합도가 만족하면 최적의 경로를 도출하였다. 최적의 경로를 도출하는 과정에서 적합도 조

건에 만족한 개체의 index를 이용하여 최적 경로 노드로 선택하였다. index를 이용한 부분은 본 논문에서 적용한 유전자 알고리즘에서 적합도가 높은 개체일수록 다음 세대를 생성할 수 있는 확률이 높기 때문이다. 따라서 개체가 도출된 것은 부모 개체가 우수한 개체이고 가장 적합한 개체가 선택된 것으로 본 논문에서는 정의하여 도출된 개체의 index를 이용하여 최적 경로를 탐색하였다. 그림 3은 단순 A*을 이용한 순차 방법과 제안된 방법, 랜덤 방법 간의 경로 탐색 결과이다.

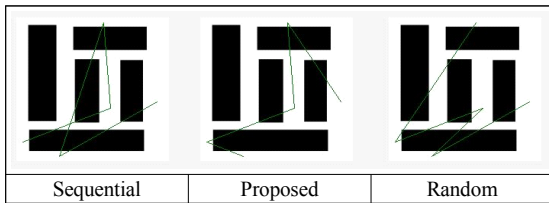


Fig. 3 Comparison of Next Node Selection Strategies

표 1은 A* 알고리즘에 적용된 거리 값을 테이블로 나타내었다.

Table. 1 Distance Matrix for A*

	index0	index1	index2	index3	index4
index0	0	175	139	35	211
index1	175	0	424	177	15
index2	139	424	0	168	279
index3	35	177	168	0	96
index4	211	15	279	96	0

Table. 2 Coordinates and Indices of Nodes

Sequential		Proposed		Random	
Index	(x, y)	Index	(x, y)	Index	(x, y)
0	184, 117	0	184, 117	0	184, 117
1	56, 193	2	114, 7	1	56, 193
2	114, 7	3	123, 126	3	123, 126
3	123, 126	4	8, 173	4	8, 173
4	8, 173	1	56, 193	2	114, 7

표 2는 A* 알고리즘을 적용하여 도출된 노드들의 좌표와 index, 제안된 방법에서 도출된 최적 경로 노드들의 좌표와 index 그리고 랜덤방법의 노드들의 좌표와

index를 나타내었다.

표 3은 A* 방법을 적용한 순차방법과 랜덤방법 및 제안한 방법 간의 총 탐색 거리를 비교하여 나타내었다. 표 3에서와 같이 제안한 방법이 A* 알고리즘을 이용한 순차 방법과 랜덤 방법보다 총 탐색 거리 값이 적어서 최적 경로 탐색에 있어서 제안된 방법이 효율적인 것을 확인할 수 있다.

Table. 3 Comparison of Total Search Distance Cost

Sequential	Proposed	Random
Index		
0 - 1 - 2 - 3 - 4	0 - 2 - 3 - 4 - 1	0 - 1 - 3 - 4 - 2
Total Fistrances		
865	419	729

VI. 결론 및 향후 개선 방향

본 논문에서는 출발지 노드부터 모든 노드들을 탐색 하면서 최적 경로를 탐색하는 방법을 제안하였다. 본 논문에서는 최적의 경로를 탐색하기 위해서 최적화 방법 중의 하나인 유전자 알고리즘을 적용하였다. 제안된 방법은 최적의 경로를 탐색하기 위해서 가중치 값으로 A* 알고리즘을 이용하여 거리를 계산하였다. 계산된 거리 값을 이용하여 개체를 생성하였고 선택 연산과 교차 연산을 적용하여 우수한 개체를 생성하였고 적합도 평가 함수를 적용하여 적합도를 계산하였다. 계산된 적합도가 조건에 만족되면 중간 경로의 노드를 도출하여 출력 노드로 선택하고, 다음 출력 노드를 선택하기 위한 출발 지점의 노드로 설정하였다. 이러한 과정을 반복하여 최적화된 출력 노드를 도출하는 효율적인 경로 탐색 방법을 제안하였다. 실험에서 알 수 있듯이 제안된 방법이 최적의 거리를 탐색하는데 있어서 효율적인 것을 확인 하였다.

향후 연구 방향은 현재에 적용된 A* 알고리즘 대신에 Deep PCM 알고리즘과 유전자 알고리즘을 적용하여 현재의 탐색 성능을 더 개선하여 내비게이션에 적용할 수 있도록 할 것이다.

REFERENCES

- [1] A. Antoniou, W. -S. Lu, *Practical Optimization : Algorithms and Engineering Applications* , Boston, MA, Springer, 2007.
- [2] K. B. Kim, D. H. Song, "Path Search Method using Genetic Algorithm," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 15, no. 6, pp. 1251-1255, Jun. 2011.
- [3] Y. G. Ryu, "Development of a shortest path searching algorithm using minimum expected weights," *The journal of The Korea Institute of Intelligent Transport Systems*, vol. 12, no 5, pp.36-45, Dec. 2013.
- [4] E. A. Hansen, S. Zilberstein, "LAO*: A heuristic search algorithm," *Artificial Intelligence* , Vol.129, pp.35-62, Mar. 2001.
- [5] K. Kim, W. S. Yang, T. S. Kim, "Optimization of Information Security Investment Portfolios Using a Genetic Algorithm," *The Journal of Korea Institute of Communications and Information Sciences*, vol.43, no.2 pp.439-451, Feb. 2018.
- [6] A. Bhardwaj , T. Manglani, "Gravitational Search Algorithm for Bidding Strategy in Uniform Price Spot Market," *International Research Journal of Engineering and Technology*, vol. 4, Issue 06, pp.5639-5644, Jun. 2017.
- [7] Y. Kim, S. Kim, "A Study on the Optimal Allocation for Intelligence Assets Using MGIS and Genetic Algorithm," *Journal of the Korean Institute of Industrial Engineers*, vol.41, no.4, pp.396-407. Aug. 2015.



김광백(Kwang Baek Kim)

1999년 부산대학교 전자계산학과 졸업(이학박사)
1997년~현재 : 신라대학교 컴퓨터소프트웨어공학부 교수
1999년~2000년 : Biomedical Fuzzy Systems Association, Editor(Japan)
2009년 : Journal of Digital Imaging (Springer: SCI), Guest Editor
2013년 : International Journal of Computational Vision and Robotics(SCOPUS), Guest Editor
2013년 : International Journal of Information and Communication Technology(SCOPUS), Guest Editor
2014년~2017 : Open Computer Science Journal, Editor,
2012년~현재 : 한국지능정보시스템학회 편집위원
2013년~현재 : International Journal of Intelligent Information Processing(SCOPUS), Editor
2015년~2016년 : Computational Intelligence and Neuroscience(SCIE), Lead Guest Editor
2016년~2017년 : 한국정보통신학회 회장
※ 관심분야 : 기계 학습, 딥 러닝, 의료 영상처리, 퍼지 논리, 유전자 알고리즘, 데이터 마이닝